EXPERIMENT 6

BLOW FISH ALGORITHM

AIM: To execute the blow fish algorithm in Python

DESCRIPTION:

Operations: (Blowfish encrypts 64-bit block with a variable length key)

1) Sub key Generation:

This process covert the key up to 448 bit long to sub keys totalling 4168 bits.

2) Data Encryption :

This process involves the iteration of a simple function 16 times. Each round contains a key dependent permutation and key and data substitution.

- Blowfish is a very fast algorithm which takes 64 bit input as plaintext and generates 64 bit output cipher text.
- It uses the concept of P-array which use of 21 bit and there are 18 P-arrays P_1 to P_{18}.
- Blowfish Algorithm runs 16 times i.e. 16 rounds

Processes:

A. Subkey Generation:

- Key Size is variable but blowfish algorithm generates very large sub-keys .The key size is in the range of 32 bits to 448 bits or 14 words.
- Concept of P-array consists of 18, 32 bit sub-keys
- There are 4 S-boxes containing 256 entries of 32 bits
- P-array is initialized first then four s boxes with fixed string
- Then P-arryas are XORed with subkeys ie from P_1 to P_{18} . Once the sub keys are generated the encryption process begins.
-

B. Data encryption and decryption:

- We use the P arrays and S boxes during this process

Algorithm for encryption of 64 bit block

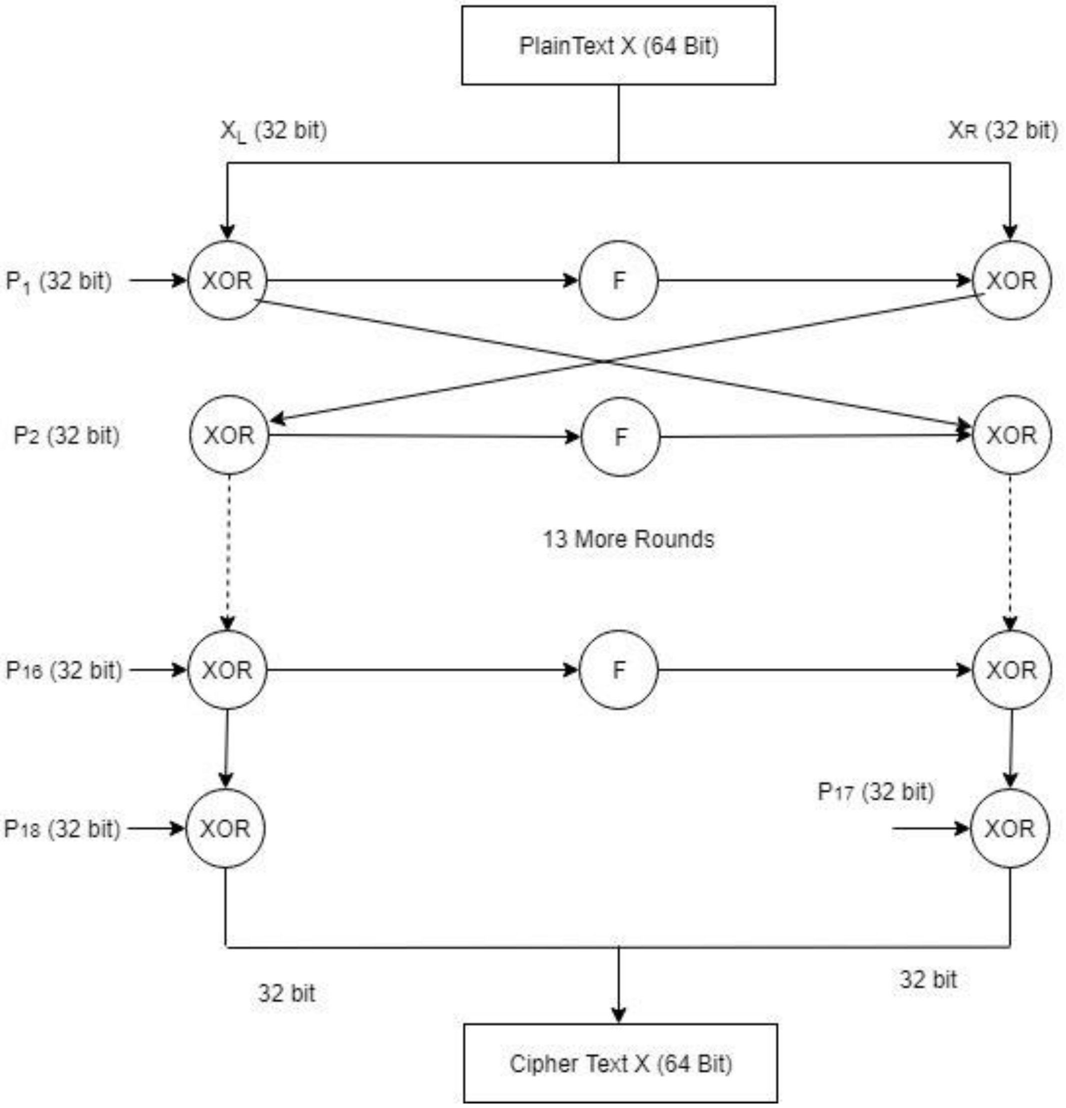
1. Divide X into two blocks CL and XR of equal sizes. Thus both XL and XR will consist of 32 bit each
2. For P=1 to 16

```
        XL = XL XOR $P_i$
        XR = f(XL) XORXR
        Swap XL ,XR
```

Next i

1. Swap XL, XR XOR P_{18}
2. XL = XL XOR P_{18}
3. XR = XR XOR P_{17}
4. Continue XL and XR back into X to get cipher text CT

Blowfish ENcryption



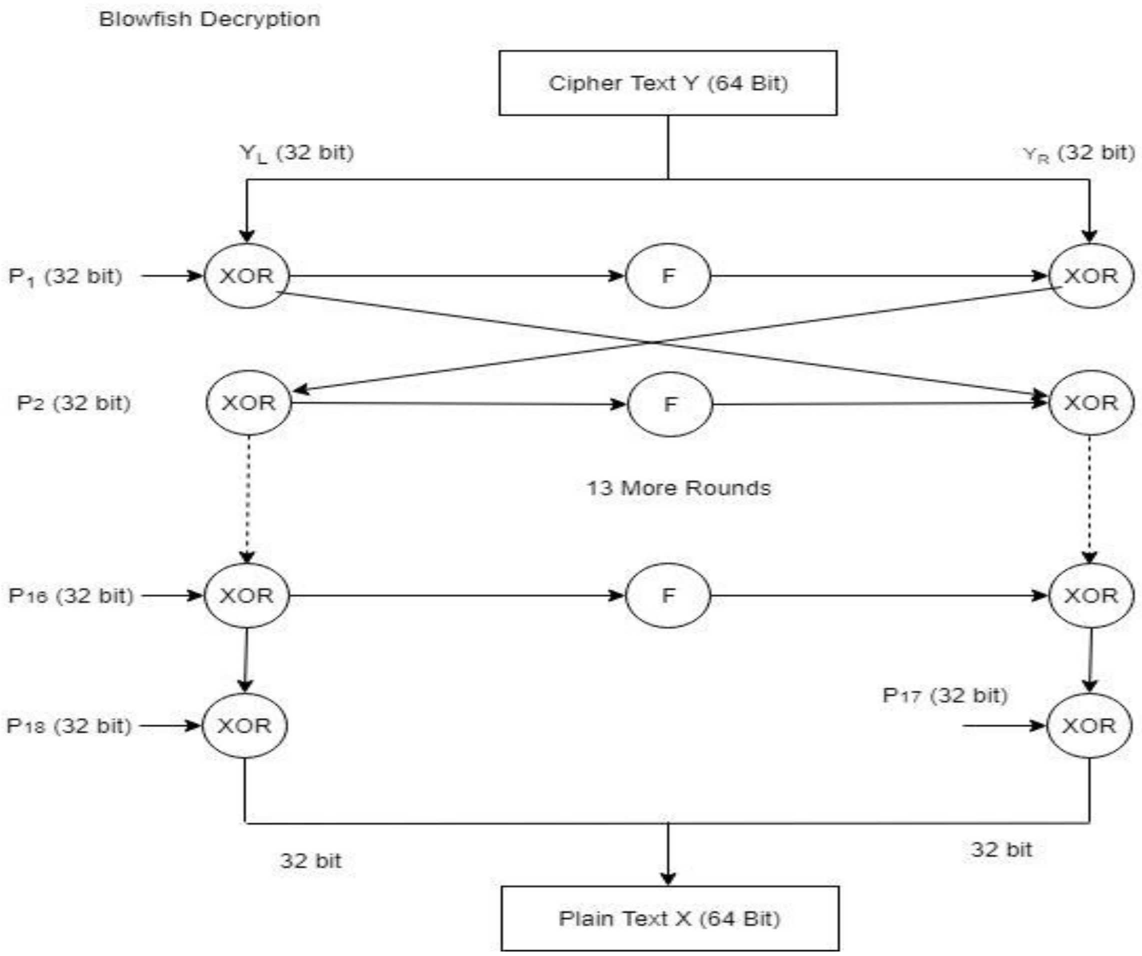
- Function f is as follows

  a. Divide the 32 bit XL block into four 8 bit sub blocks named a, b, c, d.

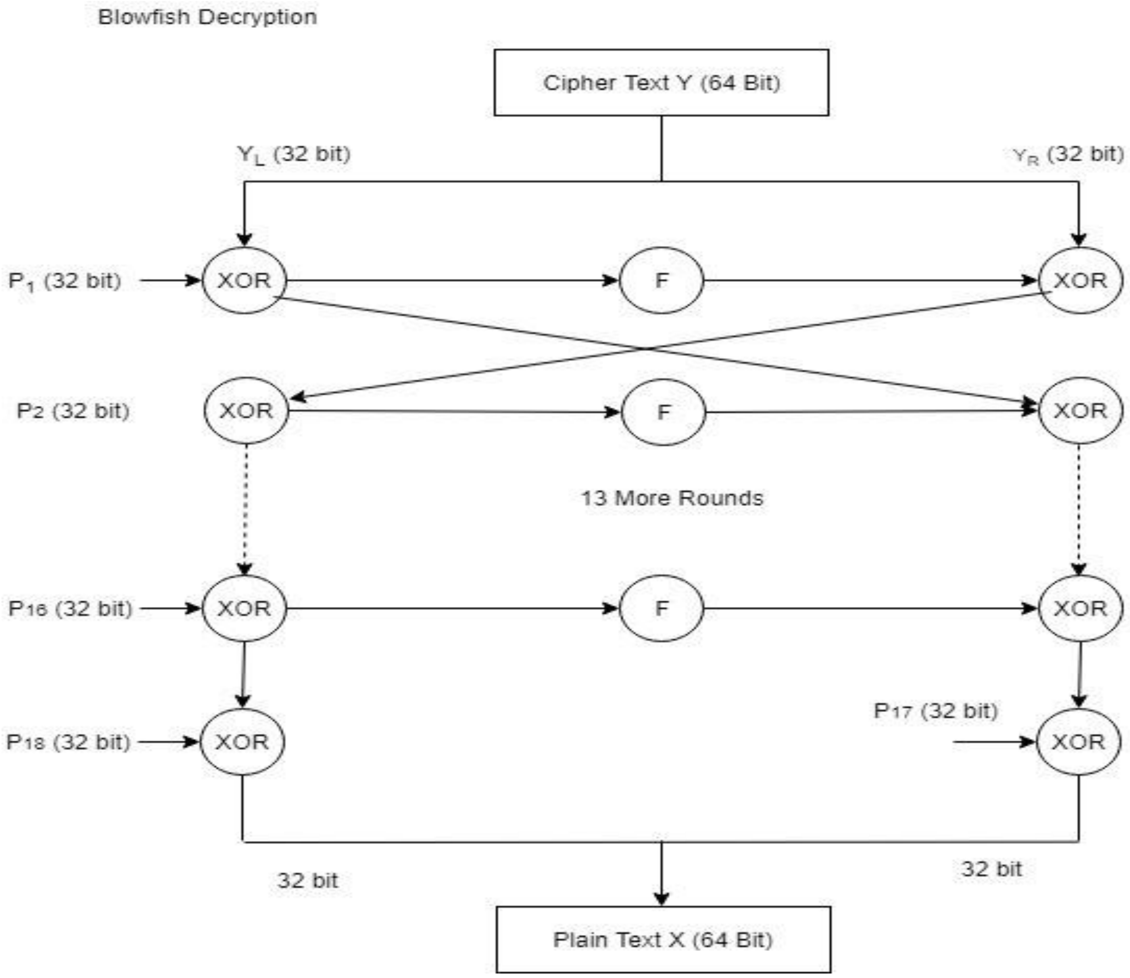
  b. Compute f(a,b,c,d) = ((S1, a + S2, b) XOR S3

  c) XORSc , d

- Function f in blowfish



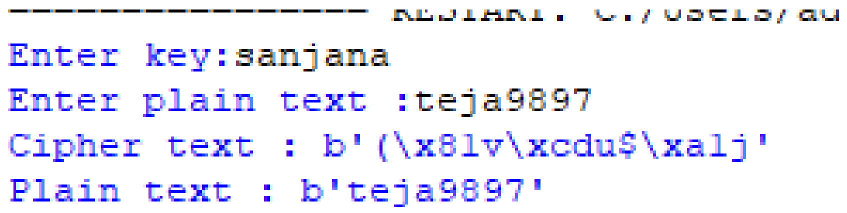
- Blowfish Decryption



CODE:

```
import blowfish
cipher=input("Enter key:")
cipher = bytes(cipher, 'utf-8')
cipher = blowfish.Cipher(cipher)
block=input("Enter plain text :")
block = bytes(block, 'utf-8')
ciphertext = cipher.encrypt_block(block)
plaintext = cipher.decrypt_block(ciphertext)
assert block == plaintext
print("Cipher text :",ciphertext)
print("Plain text :",plaintext)
```

OUTPUT:



```
================= RESTART: C:/Users/au
Enter key:sanjana
Enter plain text :teja9897
Cipher text : b'(\x81v\xcdu$\xa1j'
Plain text : b'teja9897'
```